



Integration Of Collaborative Visualization In The AccessGrid Environment

Michael Braitmaier

braitmaier@hlrs.de

University of Stuttgart, Germany

High-Performance Computing-Center Stuttgart (HLRS)

www.hlrs.de



AccessGrid Retreat: Collaborative Visualization

Slide 1

High Performance Computing Center Stuttgart

H L R I S 

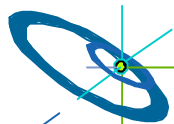
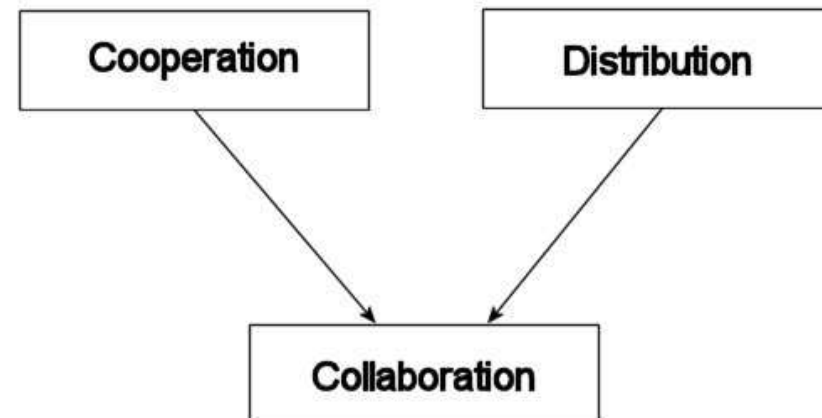
Overview

- Requirements for collaboration
- COVISE – Collaborative Visualization and Simulation Environment
- Realization of collaboration in COVISE
- The PARAVR performance analysis tool
- Technical requirements for application integration
- Generic communication library - Overview
- AccessGrid extensions
 - Hierarchical data structure
 - Extension of the VenueServer
- AccessGrid services
 - Event synchronization service
 - Whiteboard service
- Results



Requirements for Collaboration - Definitions

- Distributed work
 - Working on a project at different locations
- Cooperative work
 - Working together pursuing a common goal
- Collaborative work
 - Distributed, shared working on a workpiece with a common goal



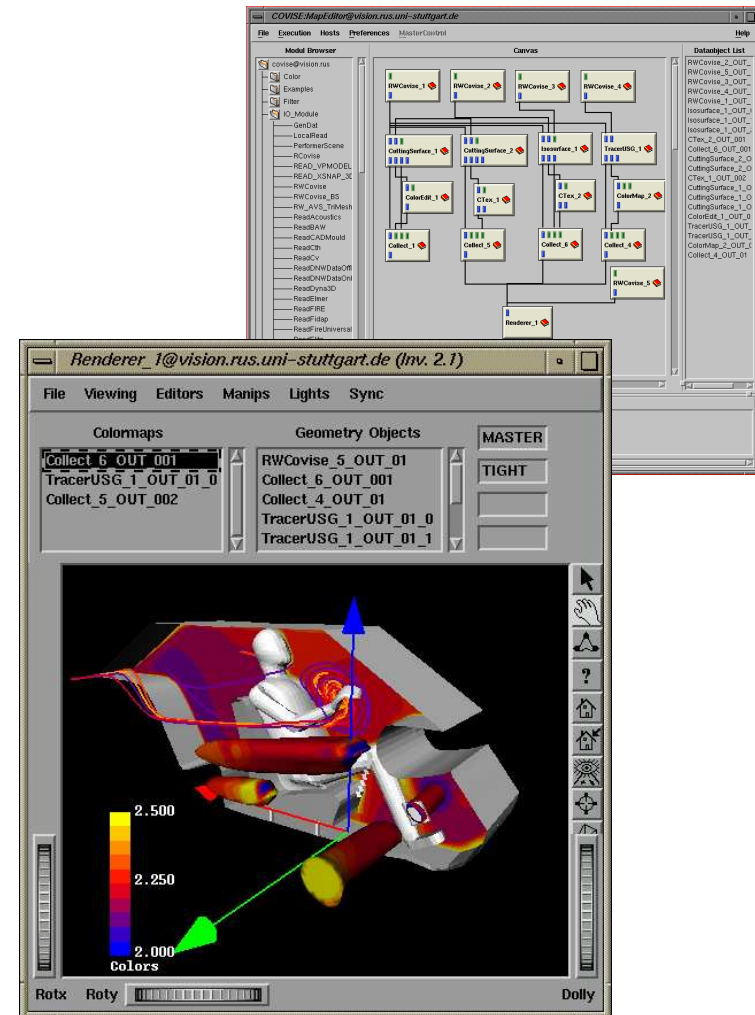
Requirements for Collaboration – Users

- Ease of use
 - Collaboration with remote sites
 - Continuous synchronization
 - Interactivity
 - Telepointing
-
- Goal: Discussion of scientific and engineering simulation results



COVISE – Collaborative Visualization and Simulation Environment

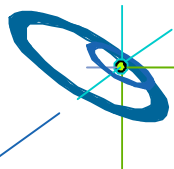
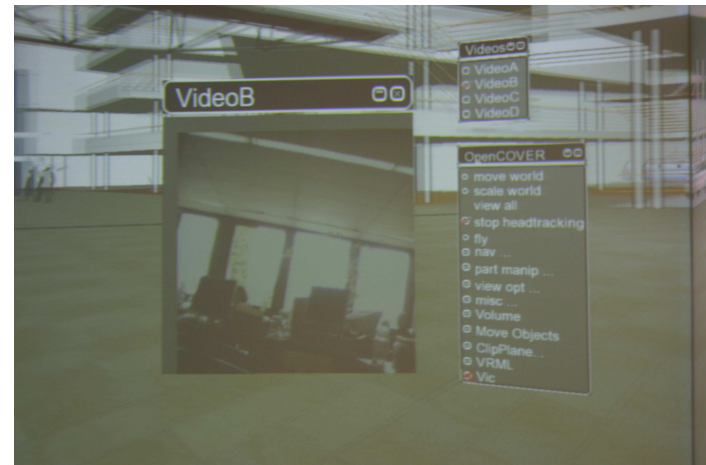
- Modular approach
- Freely configurable due to modules
- Extendable
- QT-based editor for setup of scenarios
- Useable
 - On desktop workstations
 - On Laptops
 - On Powerwalls
 - In CAVEs
- Natively collaboration aware



H L R I S

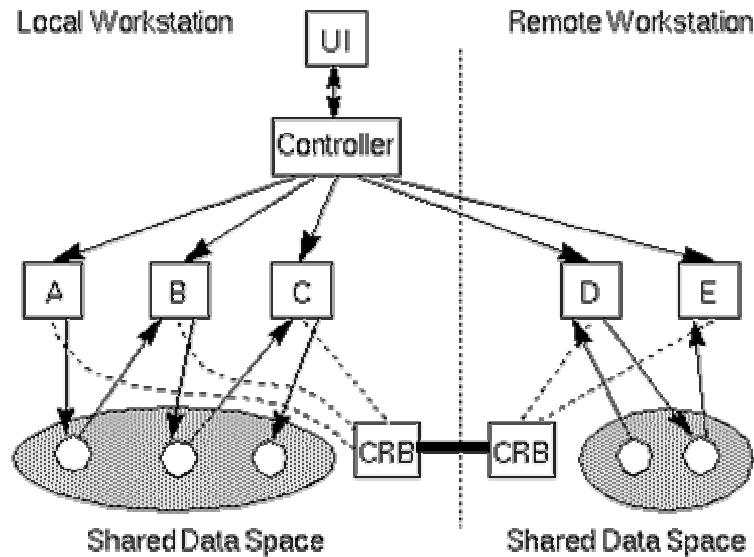
COVISE Collaboration features

- Session management
- Avatar
- Data distribution through CoviseRequestBroker (CRB)
- Event synchronization through unique Covise Controller
- Local data
- Low bandwidth
- Use of tangible interface and augmented reality
- Requirements for AccessGrid
 - Integration into AG
 - Integration of AG into COVISE
 - Intuitive GUI for AG control

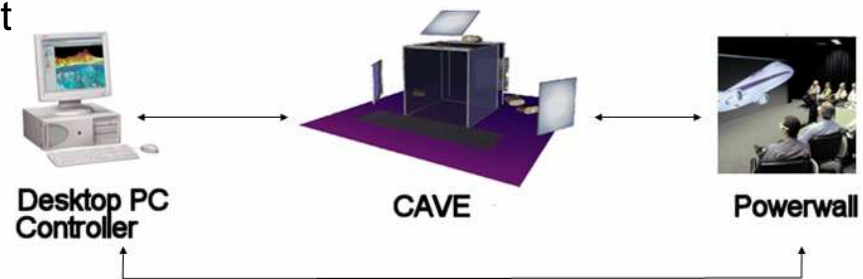


Realization of Collaboration in COVISE

- Concept of COVISE

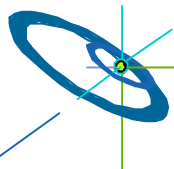
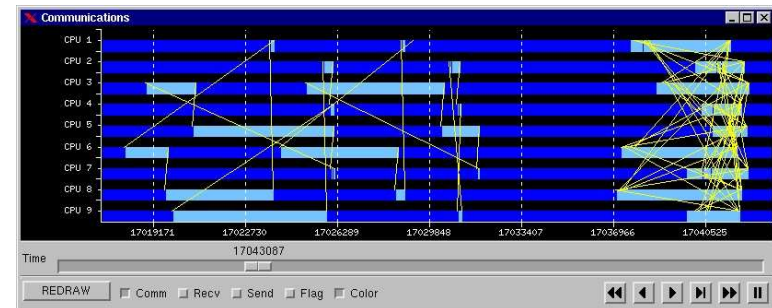
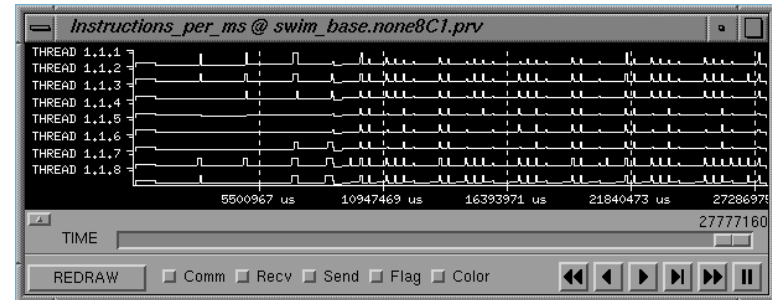


- Central synchronization object (Controller)
- Based on sockets



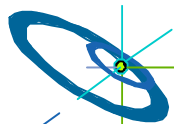
PARAVER performance analysis tool

- Visualization and performance analysis tools
- Analyze MPI, OpenMP, Java, ...
- Uses different visualization modes
- Distribute analysis data before visualization
- Requirements for Collaboration
 - Distribute analysis data
 - Synchronize instances
- Current approach: VNC



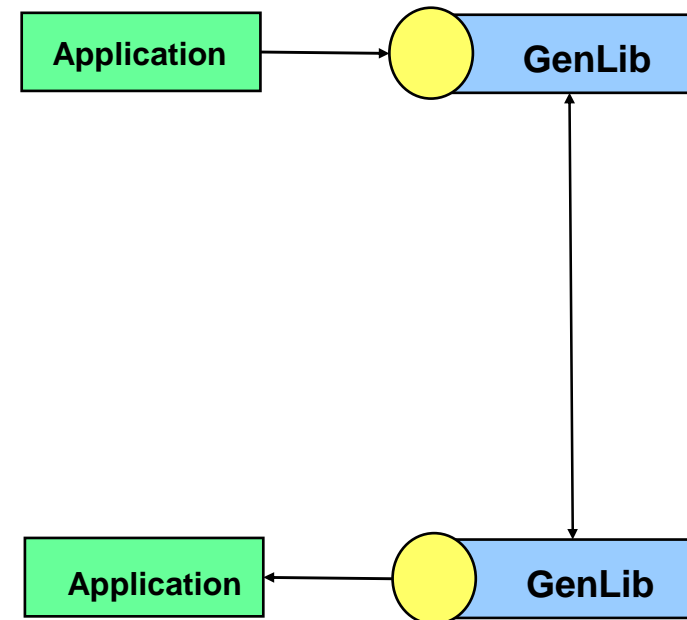
Technical requirements for application integration

- Collaboration unaware applications: PARAVR
 - Event synchronization mechanism
→ Generic Communication Library
 - Propagate whole state of application → higher bandwidth
 - Need initial distribution of trace files and data files
→ Shared data space
- Collaboration aware applications
 - COVISE
 - **Make Controller known → store Controller server IP**
 - **Propagate connect information of Controller to clients**
 - MAF
 - **Analogous to COVISE**
 - **Can run on the Generic Communication Library**



Generic Communication Library - Overview

- Event synchronization
- Goal: easy integration
- Session management
 - User management
 - Rights management?
- Firewall issues



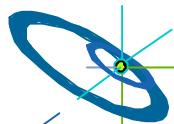
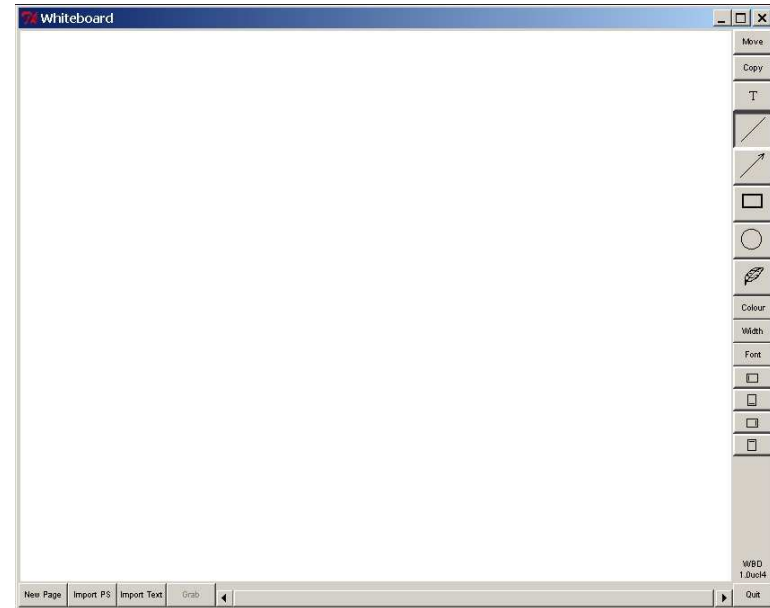
Required AccessGrid extensions

- Server side changes
 - Storage of IP addresses
 - **Extend the Venue object by an additional data element**
 - **Adjust VenueManagement tool to allow addition of values**
- Service development
 - Event synchronization service for COVISE
 - **Used for making the Controller known to all entering participants**
 - Shared whiteboard service using wbd
- Hierarchical data structure
 - Extend existing object structure to keep changes low
 - Virtual directories
 - Upload of multiple objects
 - File access from client-side



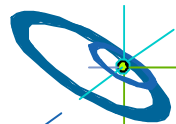
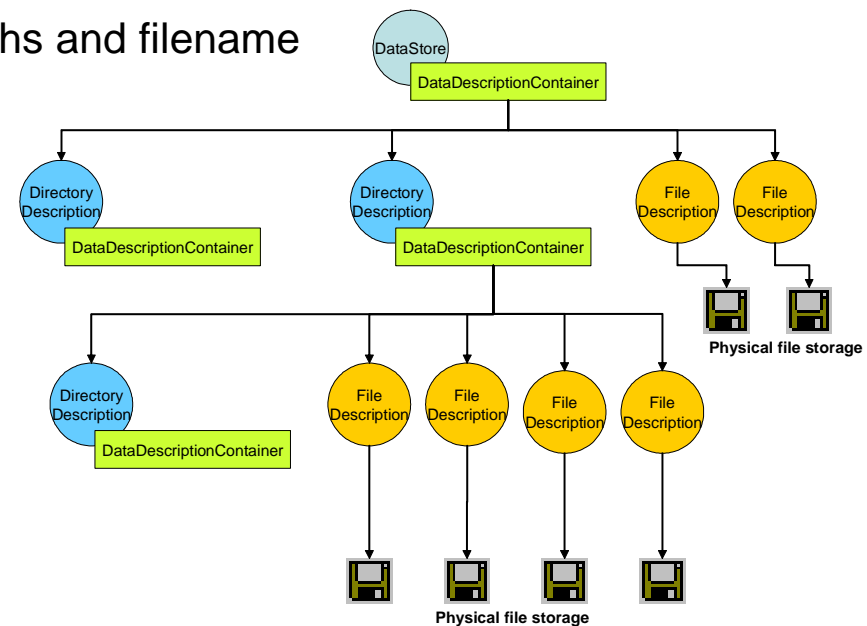
Services

- Whiteboard service
 - Analogous to rat
 - Service for wbd
 - Service for mlb
 - Modify mlb to connect to eBeam whiteboard system
- COVISE service
 - Determine Controller server connection data
 - **Adapted stream configuration**
 - Startup local application
 - Possibly start controller when none available
 - enter Controller IP in Venue? (currently static)



Hierarchical data storage – design ideas

- Introduce DirectoryDescription
- Usage of DataDescriptionContainer for creation of hierarchy
- Directory virtual on server-side
- Storage on server stays the same
- Access files through virtual paths and filename

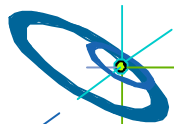
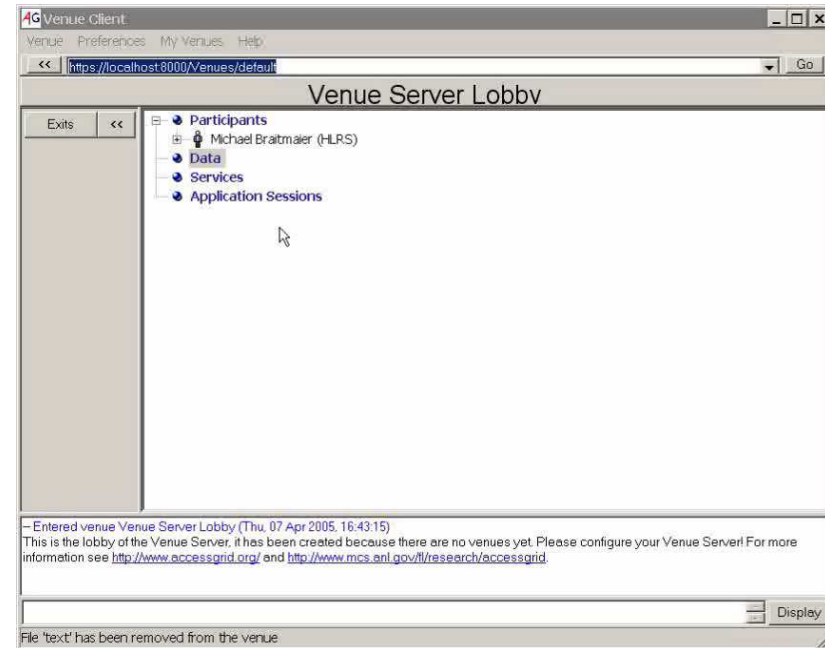


Results – Hierarchical data structure

- Directories
 - Add new
 - Remove
 - Persistent while server is up
- Files
 - Add to directories
 - Remove from directories

Constraints so far:

- No multi-add so far
- No complete directory-add so far
- Not yet persistent when VenueServer restarts

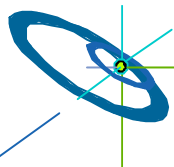
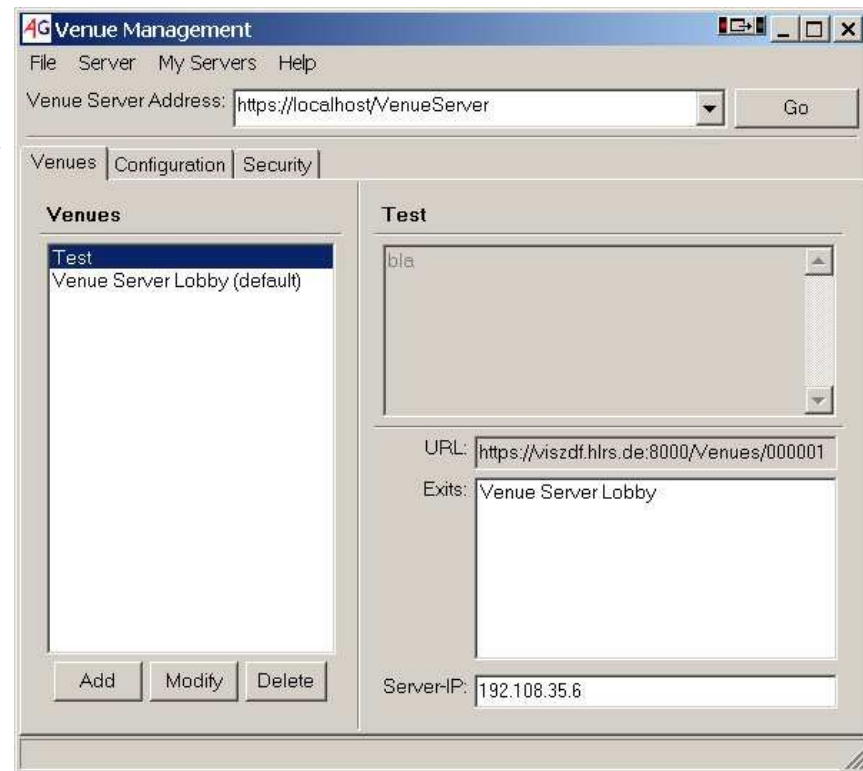


Results – Venue extension and services

- Venue extended by additional value
- Adapted VenueManagement tool
- Missing possibility to enter multiple addresses
- Desired: Identifier,address -pair

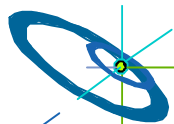
Services:

- Wbd service
- Mlb service



Summary

- Integration of several applications into AG
 - Collaboration aware and unaware
- Requirements for successful integration
 - Hierarchical data storage in AG
 - Additional services
 - Communication library for synchronization and session management services
 - GUI adaption to new requirements
- Supporting tools and extensions for AG
 - Services, SharedApps, GUI-Tools
- Results
 - Hierarchical data storage prototypic version
 - Venue IP-address storage
 - Whiteboard services
 - Communication library in development
 - Support for physical whiteboards in progress



Thank you for your attention

- Questions?
- Comments?

